Detecting Early Stage Cancer using Liquid Biopsy

Jackie Baek Vivek Farias Chinmay Jha Andrew Li **Deeksha Sinha**
MIT

Motivation

Motivation

- Successful cancer treatment requires early detection

Motivation

- Successful cancer treatment requires early detection
- Early cancer detection remains an open problem

Motivation

- Successful cancer treatment requires early detection
- Early cancer detection remains an open problem
  - Relies on invasive biopsies

Motivation

- Successful cancer treatment requires early detection
- Early cancer detection remains an open problem
  - Relies on invasive biopsies
- *Liquid biopsy*: a simple, non-invasive blood test

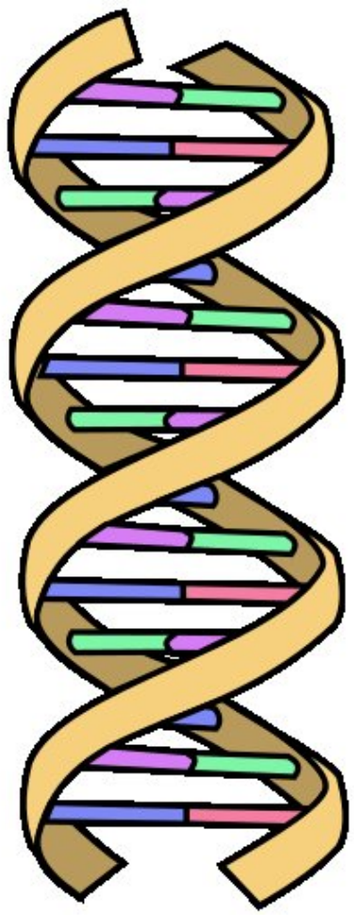
Motivation

- Successful cancer treatment requires early detection
- Early cancer detection remains an open problem
  - Relies on invasive biopsies
- *Liquid biopsy*: a simple, non-invasive blood test
  - Fueled by tremendous reductions in DNA sequencing cost

Motivation

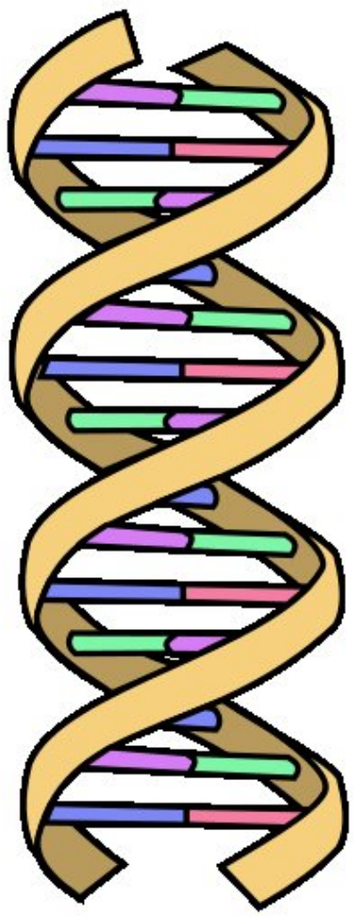
- Successful cancer treatment requires early detection
- Early cancer detection remains an open problem
  - Relies on invasive biopsies
- *Liquid biopsy*: a simple, non-invasive blood test
  - Fueled by tremendous reductions in DNA sequencing cost
- **Goal:** develop a cost-efficient liquid biopsy to detect early-stage cancer

Biology 101



DNA

Biology 101

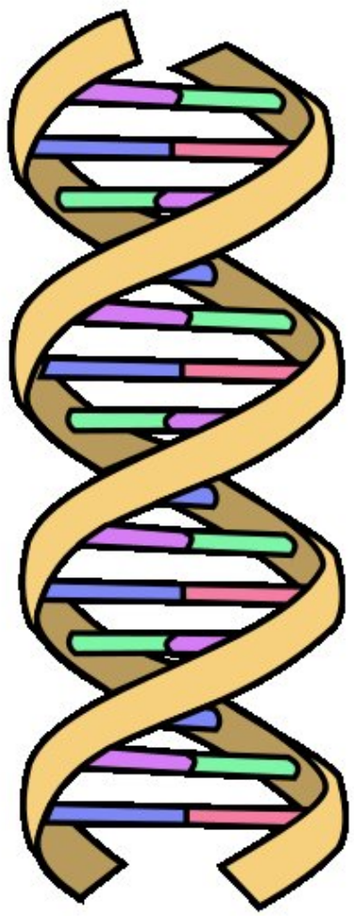


DNA

- DNA can be represented as a string: $\{A,C,G,T\}^{3\text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

Biology 101



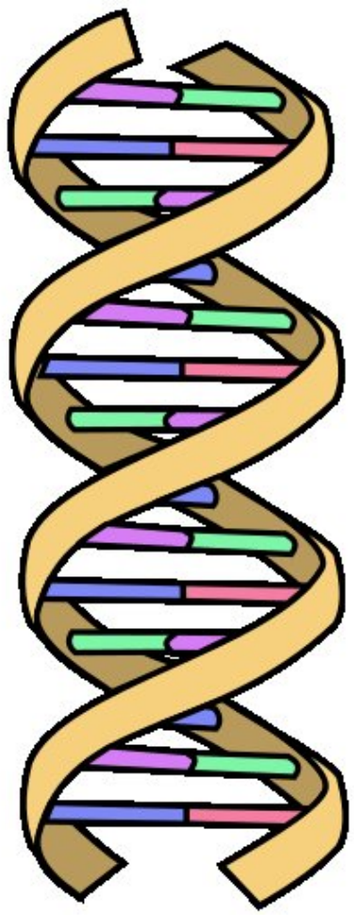
DNA

- DNA can be represented as a string: {A,C,G,T}$^{3\ billion}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

- An **address** is a specific position in this string

Biology 101



DNA

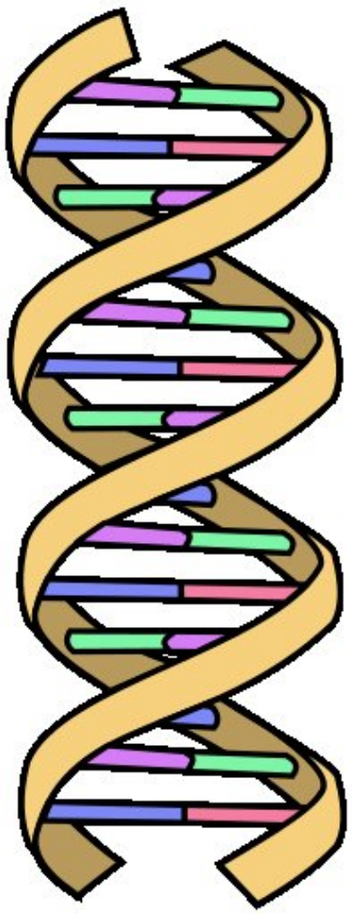
- DNA can be represented as a string: {A,C,G,T}$^{3 \text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

gene

- An **address** is a specific position in this string
- A **gene** is a substring which encodes instructions to create a protein (of length ~10,000)

Biology 101



DNA

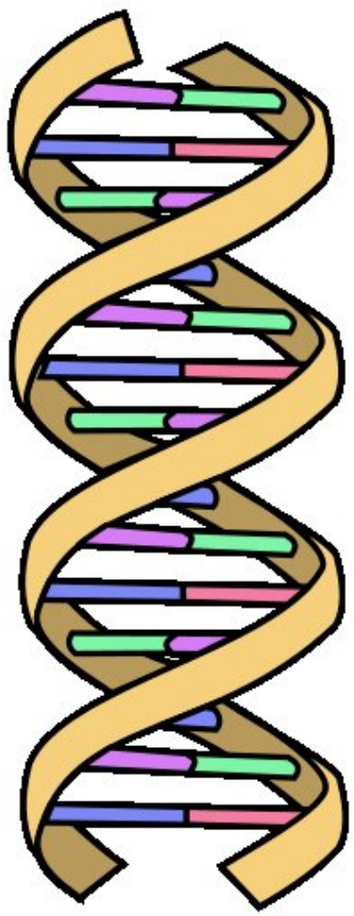
- DNA can be represented as a string: {A,C,G,T}$^{3 \text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

gene

- An **address** is a specific position in this string
- A **gene** is a substring which encodes instructions to create a protein (of length ~10,000)
- A **mutation** is a change in your DNA (insertion, deletion, etc.)

Biology 101



DNA

- DNA can be represented as a string: {A,C,G,T}$^{3 \text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

gene

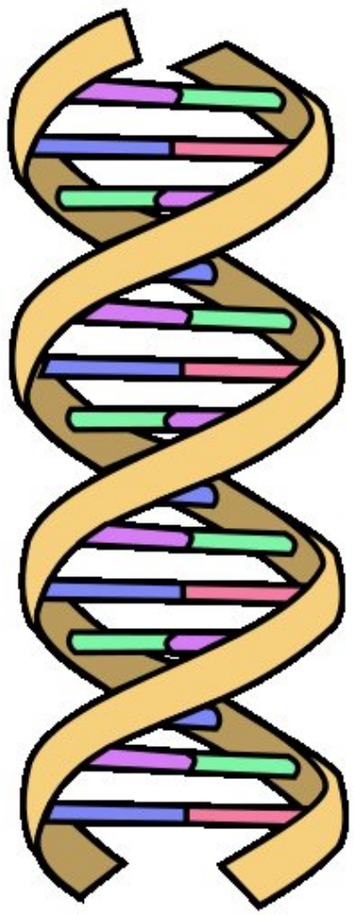
- An **address** is a specific position in this string
- A **gene** is a substring which encodes instructions to create a protein (of length ~10,000)
- A **mutation** is a change in your DNA (insertion, deletion, etc.)

...ATGC...

...ATGAC...

insertion

Biology 101



DNA

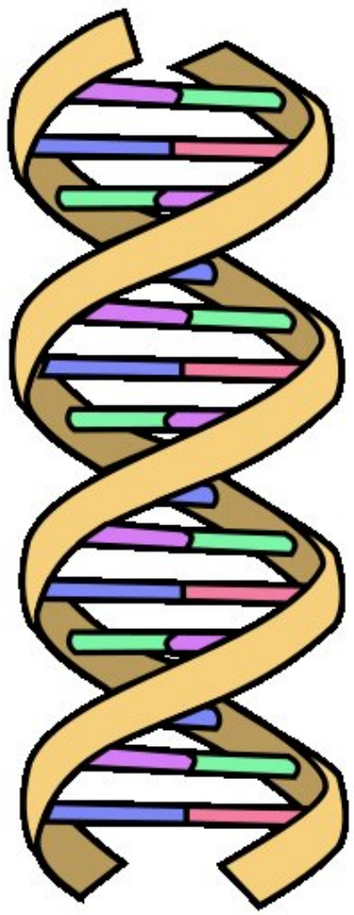
- DNA can be represented as a string: {A,C,G,T}$^{3 \text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

gene

- An **address** is a specific position in this string
- A **gene** is a substring which encodes instructions to create a protein (of length ~10,000)
- A **mutation** is a change in your DNA (insertion, deletion, etc.)
- **Cancer** is caused by mutations that cause abnormal cell growth

Biology 101



DNA

- DNA can be represented as a string: {A,C,G,T}$^{3 \text{ billion}}$

...AGCATGCAGTACGTACGTCACATTCGATCGATGG...

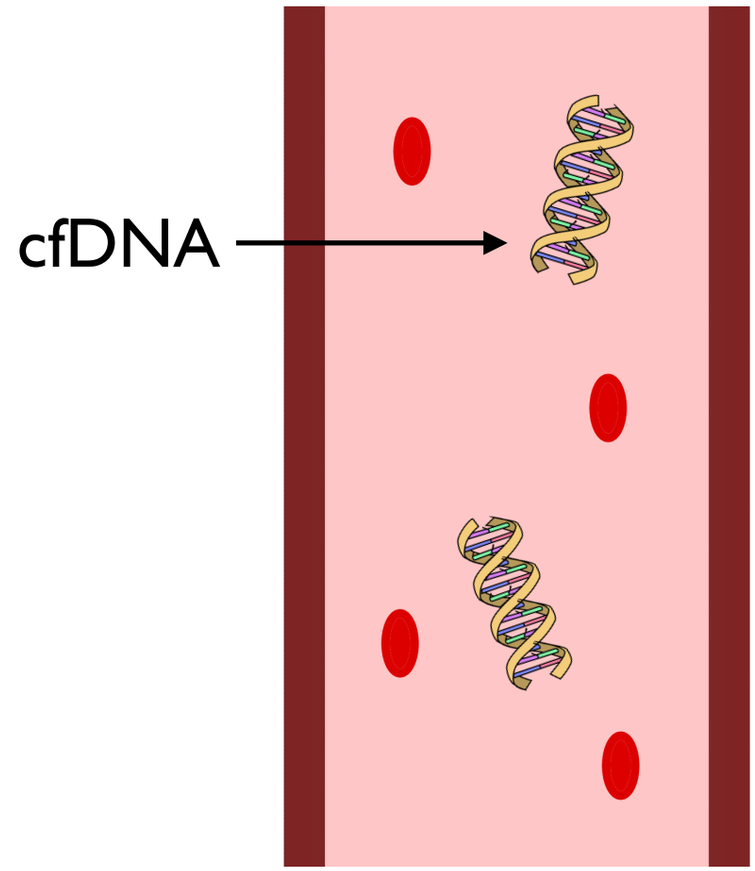
gene

- An **address** is a specific position in this string
- A **gene** is a substring which encodes instructions to create a protein (of length ~10,000)
- A **mutation** is a change in your DNA (insertion, deletion, etc.)
- **Cancer** is caused by mutations that cause abnormal cell growth
- **Sequencing** is the process of recovering the DNA string

Detecting Early-Stage Cancer

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood



Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations

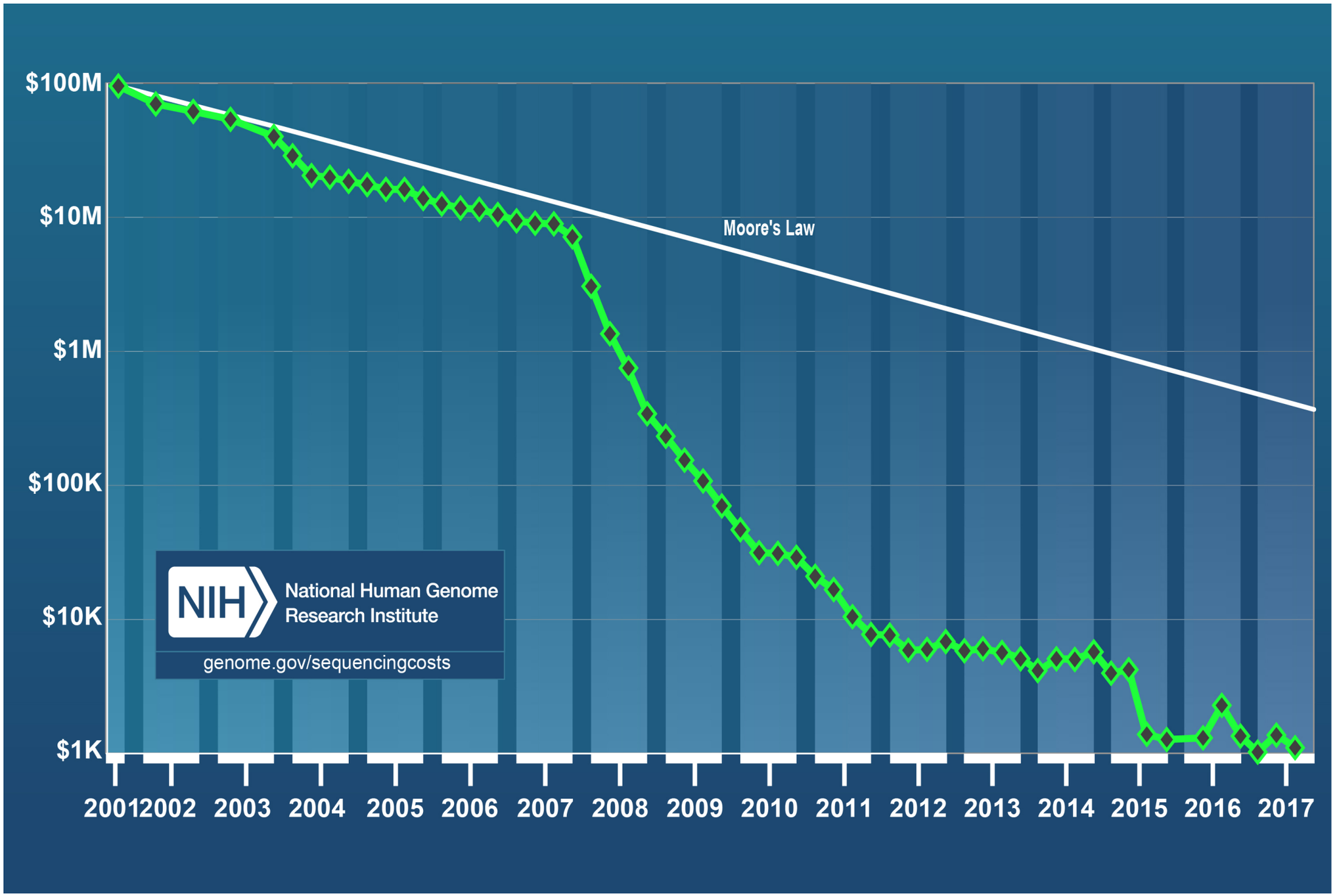
Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations
3. Predict cancer using mutation information

Cost of Sequencing a Human Genome



Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations
3. Predict cancer using mutation information

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations
3. Predict cancer using mutation information

- **Cost of test:** cost of sequencing x 10,000

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations
3. Predict cancer using mutation information

- **Cost of test:** cost of sequencing x 10,000
- **Opportunity:** Don't need to sequence the *entire* 3 billion length DNA

AmpliSeq™ for Illumina Cancer Hotspot Panel v2

Fast, accurate investigation of hotspot regions in 50 genes with known cancer associations.

Highlights

- **Relevant Gene Content**
  Target ~2800 COSMIC mutations from 50 oncogenes and tumor suppressor genes
- **Fast, Streamlined Workflow**
  Prepare sequencing-ready libraries in a single day from as little as 1 ng high-quality DNA or 10 ng DNA from FFPE tissue
- **Accurate Data**
  Detect somatic mutations down to 5% frequency using local or cloud-based analysis

Introduction

The AmpliSeq for Illumina Cancer Hotspot Panel v2 is a targeted resequencing assay for researching somatic mutations across the hotspot regions of 50 genes with known associations to cancer

Table 1: AmpliSeq for Illumina Cancer Hotspot Panel v2

Parameter	Specification
No. of Genes	50
Targets	Hotspot regions within oncogenes and tumor suppressor genes
Cumulative Target Size	22 kb
Variant Types	SNVs, indels[a]
Amplicon Size	106 bp on average
No. of Amplicons	207
Input DNA Requirement	1–100 ng (10 ng recommended)
No. of Pools per Panel	1
Supported Sample Types	FFPE tissue, blood
Percent Targets Covered at Minimum 500× at Recommended Throughput	> 95%
Coverage Uniformity (percent of targets with > 0.2× mean coverage)	> 95%
Percent On-Target Aligned Reads	> 80%
Total Assay Time	5 hours[b]

AmpliSeq™ for Illumina Cancer Hotspot Panel v2

Fast, accurate investigation of hotspot regions in 50 genes with known cancer associations.

Highlights

- **Relevant Gene Content**
  Target ~2800 COSMIC mutations from 50 oncogenes and tumor suppressor genes
- **Fast, Streamlined Workflow**
  Prepare sequencing-ready libraries in a single day from as little as 1 ng high-quality DNA or 10 ng DNA from FFPE tissue
- **Accurate Data**
  Detect somatic mutations down to 5% frequency using local or cloud-based analysis

Introduction

The AmpliSeq for Illumina Cancer Hotspot Panel v2 is a targeted resequencing assay for researching somatic mutations across the hotspot regions of 50 genes with known associations to cancer

Table 1: AmpliSeq for Illumina Cancer Hotspot Panel v2

Parameter	Specification
No. of Genes	50
Targets	Hotspot regions within oncogenes and tumor suppressor genes
Cumulative Target Size	22 kb
Variant Types	SNVs, indels[a]
Amplicon Size	106 bp on average
No. of Amplicons	207
Input DNA Requirement	1–100 ng (10 ng recommended)
No. of Pools per Panel	1
Supported Sample Types	FFPE tissue, blood
Percent Targets Covered at Minimum 500× at Recommended Throughput	> 95%
Coverage Uniformity (percent of targets with > 0.2× mean coverage)	> 95%
Percent On-Target Aligned Reads	> 80%
Total Assay Time	5 hours[b]

Detecting Early-Stage Cancer

- **Cell-free DNA (cfDNA):** DNA that flows freely in the blood

New test for cancer detection (Heitzer et al., Wan et al.)

1. Perform blood test ("liquid biopsy")
2. Repeat 10,000 times:
    a. Look for cfDNA in the blood
    b. Sequence DNA and look for mutations
3. Predict cancer using mutation information

- **Cost of test:** cost of sequencing x 10,000
- **Opportunity:** Don't need to sequence the *entire* 3 billion length DNA

Which parts of the DNA should be sequenced to detect cancer and classify its type at the lowest possible cost?

Model

Model

Data

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{25000}$$

$$Y = \begin{bmatrix} \text{liver} \\ \text{stomach} \\ \text{lung} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{25000} \qquad Y = \begin{bmatrix} \text{liver} \\ \text{stomach} \\ \text{lung} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

- Focus on two-class cancer type classification for this talk

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{4000} \qquad Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

- Focus on two-class cancer type classification for this talk

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{4000} \qquad Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

Challenges

- Extremely sparse: 0.03%
- High-dimensional data with limited samples

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{4000} \qquad Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

Cost Model

Cost of sequencing = (# of addresses) + 42 x (# intervals)

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

3 billion

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}$$

4000

$$Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

Cost Model

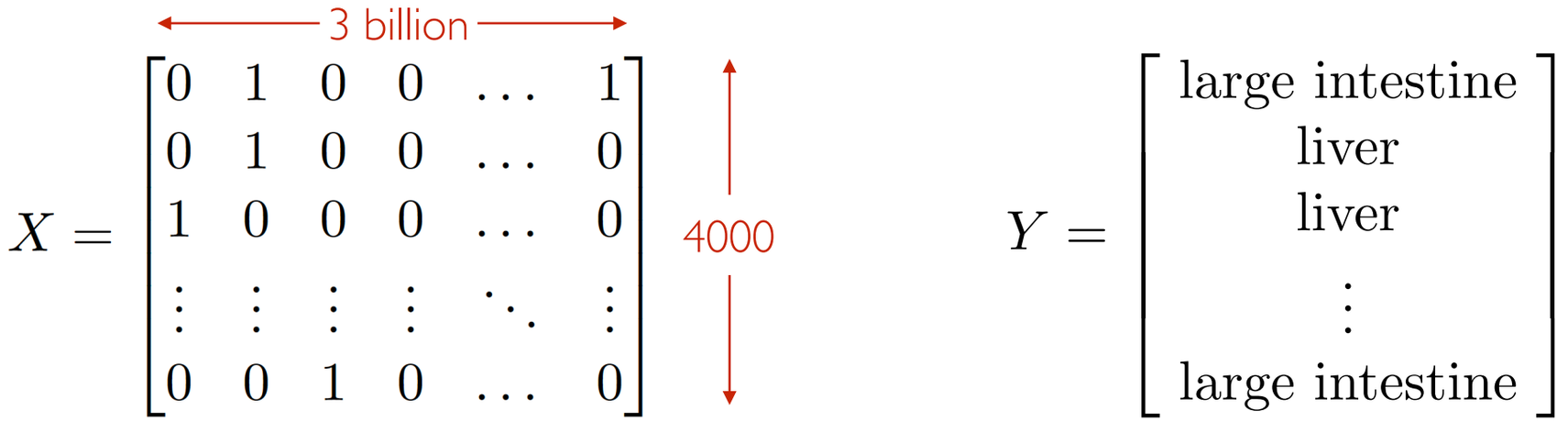
Cost of sequencing = (# of addresses) + 42 x (# intervals)

...AGCATGCAGTACGTACGTCACATTCGATCGATGGTACGTCGTA...

Model

Data

COSMIC database: genome-wide mutation data from cancer patients



Cost Model

Cost of sequencing = (# of addresses) + 42 x (# intervals)

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \quad Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

3 billion

4000

Cost Model

Cost of sequencing = (# of addresses) + 42 x (# intervals)

= (8 + 11) + (42 x 2)

8 11

...AGCATGCAGTACGTACGTCACATTCGATCGATGGTACGTCGTA...

Model

Data

COSMIC database: genome-wide mutation data from cancer patients

$$X = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}^{\text{3 billion}} \Bigg\} \text{4000} \qquad Y = \begin{bmatrix} \text{large intestine} \\ \text{liver} \\ \text{liver} \\ \vdots \\ \text{large intestine} \end{bmatrix}$$

Cost Model

Cost of sequencing = (# of addresses) + 42 x (# intervals)

Question

Given a cost budget, what is the best classifier?

Classification Model Features

Classification Model Features

- Mutation in each address in the DNA

Classification Model Features

- Mutation in each address in the DNA
- New features:

Classification Model Features

- Mutation in each address in the DNA
- New features:
  - Mutation anywhere in the gene can impact protein

Classification Model Features

- Mutation in each address in the DNA
- New features:
  - Mutation anywhere in the gene can impact protein
  - Look for predictive regions in the gene

Classification Model Features

- Mutation in each address in the DNA
- New features:
  - Mutation anywhere in the gene can impact protein
  - Look for predictive regions in the gene
  - Add features representing a mutation in a **region** in the gene

New features

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 \end{bmatrix}$$

New features

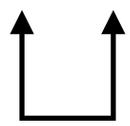
$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 \end{bmatrix}$$

- Add feature representing a mutation in either of the first two addresses

New features

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 1 & 0 & \dots & 1 & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

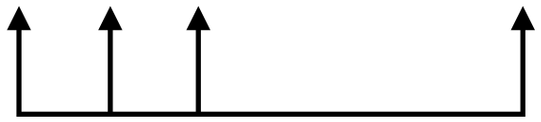


$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 & 44 \end{bmatrix}$$

- Add feature representing a mutation in either of the first two addresses

New features

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & \dots & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & \dots & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$



$$Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 & 44 & 88 \end{bmatrix}$$

- Add feature representing a mutation in either the first, second, third or the last address

New features

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 & 1 & 1 & \dots \\ 0 & 0 & 1 & 0 & \dots & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & \dots & 0 & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & \dots \end{bmatrix} \qquad Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 & 44 & 88 & \dots \end{bmatrix}$$

- Add features corresponding to all subsets of addresses within a gene

New features

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 1 & 1 & 1 & \dots \\ 0 & 0 & 1 & 0 & \dots & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & \dots & 0 & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & \dots \end{bmatrix} \qquad Y = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$C = \begin{bmatrix} 43 & 43 & 43 & 43 & \dots & 43 & 44 & 88 & \dots \end{bmatrix}$$

- Add features corresponding to all subsets of addresses within a gene
- Exponential increase in number of features

CSVM

$$\min_{\theta} \sum_{i=1}^{m} \mathbf{I}\{f_\theta(x_i) \neq y_i\}$$

$$\text{s.t. cost of classifier} \leq \text{budget}$$

CSVM

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t. cost of classifier} \leq \text{budget}$$

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint
- Use column generation:

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features
  - Repeat:

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features
  - Repeat:
    - Find optimal classifier over current set of features

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features
  - Repeat:
    - Find optimal classifier over current set of features
    - Add feature which further reduces misclassification loss

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

Dual

$$\max_{w,v} \sum_{i=1}^{n} e^T w - \gamma v$$

$$\text{s.t.} \left| \sum_{i=1}^{m} w_i X_i^j y_i \right| \leq c_j v, j \in [n],$$

$$w^T y = 0,$$

$$0 \leq w \leq 1.$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features
  - Repeat:
    - Find optimal classifier over current set of features
    - Add feature which further reduces misclassification loss

CSVM

$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t.} \sum_{j=1}^{n} c_j |a_j| \leq \gamma$$

Dual

$$\max_{w,v} \sum_{i=1}^{n} e^T w - \gamma v$$

$$\text{s.t.} \left| \sum_{i=1}^{m} w_i X_i^j y_i \right| \leq c_j v, j \in [n],$$

$$w^T y = 0,$$

$$0 \leq w \leq 1.$$

- Classifier cost encoded as a linear constraint
- Use column generation:
  - Initialize with a small set of features
  - Repeat:
    - Find optimal classifier over current set of features
    - Add feature which further reduces misclassification loss

Column generation

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)
- $\left|y^T X^j\right|$ is imbalance in number of mutations across the two classes

$$X^j = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad \left|y^T X^j\right| = 3 - 1$$

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)
- $\left|y^T X^j\right|$ is imbalance in number of mutations across the two classes
- $\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|$ is weighted imbalance in the feature

$$X^j = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad \left|y^T X^j\right| = 3 - 1$$

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)
- $\left|y^T X^j\right|$ is imbalance in number of mutations across the two classes
- $\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|$ is weighted imbalance in the feature
- Identify feature with highest imbalance to cost ratio

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)
- $\left|y^T X^j\right|$ is imbalance in number of mutations across the two classes
- $\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|$ is weighted imbalance in the feature
- Identify feature with highest imbalance to cost ratio

$$\max_{j \in [n]} \frac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j}$$

Column generation

- Find feature j such that $\dfrac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j} > v$
- Dual variable w encodes sample relevance (weight)
- $\left|y^T X^j\right|$ is imbalance in number of mutations across the two classes
- $\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|$ is weighted imbalance in the feature
- Identify feature with highest imbalance to cost ratio

$$\max_{j \in [n]} \frac{\left|\sum_{i=1}^{m} w_i X_i^j y_i\right|}{c_j}$$

Challenging optimization problem because of large number of features

Column generation

$$\max_{j \in [n]} \frac{\left| \sum_{i=1}^{m} w_i X_i^j y_i \right|}{c_j}$$

Column generation

$$\max_{j \in [n]} \frac{\left| \sum_{i=1}^{m} w_i X_i^j y_i \right|}{c_j}$$

- Decompose optimization problem over genes

Column generation

$$\max_{g} \max_{j \in [n_g]} \frac{\left| \sum_{i=1}^{m} w_i X_i^j y_i \right|}{c_j}$$

Column generation

$$\max_{g} \max_{j \in [n_g]} \frac{\left| \sum_{i=1}^{m} w_i X_i^j y_i \right|}{c_j}$$

- Focus on the inner maximization problem and then solve it over all the genes

Column generation

$$\max_{j \in [n_g]} \frac{\left| \sum_{i=1}^{m} w_i X_i^j y_i \right|}{c_j}$$

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

- Find feature with the maximum number of mutations across the given samples

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

- Find feature with the maximum number of mutations across the given samples
- Feature is made from a collection of addresses

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

- Find feature with the maximum number of mutations across the given samples
- Feature is made from a collection of addresses
- Each address ‘covers’ a set of samples: samples which have mutations in that address

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

- Find feature with the maximum number of mutations across the given samples
- Feature is made from a collection of addresses
- Each address ‘covers’ a set of samples: samples which have mutations in that address
- Each feature represents set of samples covered by any of the addresses in the feature

Column generation

$$\max_{j \in [n_g]} \sum_{i=1}^{m} X_i^j$$

- Find feature with the maximum number of mutations across the given samples
- Feature is made from a collection of addresses
- Each address ‘covers’ a set of samples: samples which have mutations in that address
- Each feature represents set of samples covered by any of the addresses in the feature

Column generation

Column generation

$$\max e^T t$$
$$\text{s.t.} \sum_{a \in A_i} u_a \geq t_i \quad \forall i \in [m]$$
$$u, t \in \{0, 1\}$$

- $t_i = 1$ if sample i is covered
- $u_a = 1$ if address a is chosen

Column generation

$$\max\ e^T t$$
$$\text{s.t.}\ \sum_{a \in A_i} u_a \geq t_i \quad \forall i \in [m]$$
$$u, t \in \{0, 1\}$$

- $t_i = 1$ if sample i is covered
- $u_a = 1$ if address a is chosen
- #variables = #samples + #addresses

Column generation

$$\max e^T t$$
$$\text{s.t.} \sum_{a \in A_i} u_a \geq t_i \quad \forall i \in [m]$$
$$u, t \in \{0, 1\}$$

- $t_i = 1$ if sample i is covered
- $u_a = 1$ if address a is chosen
- #variables = #samples + #addresses
    - Significant reduction in problem size

Column generation

$$\max e^T t$$
$$\text{s.t.} \sum_{a \in A_i} u_a \geq t_i \quad \forall i \in [m]$$
$$u, t \in \{0, 1\}$$

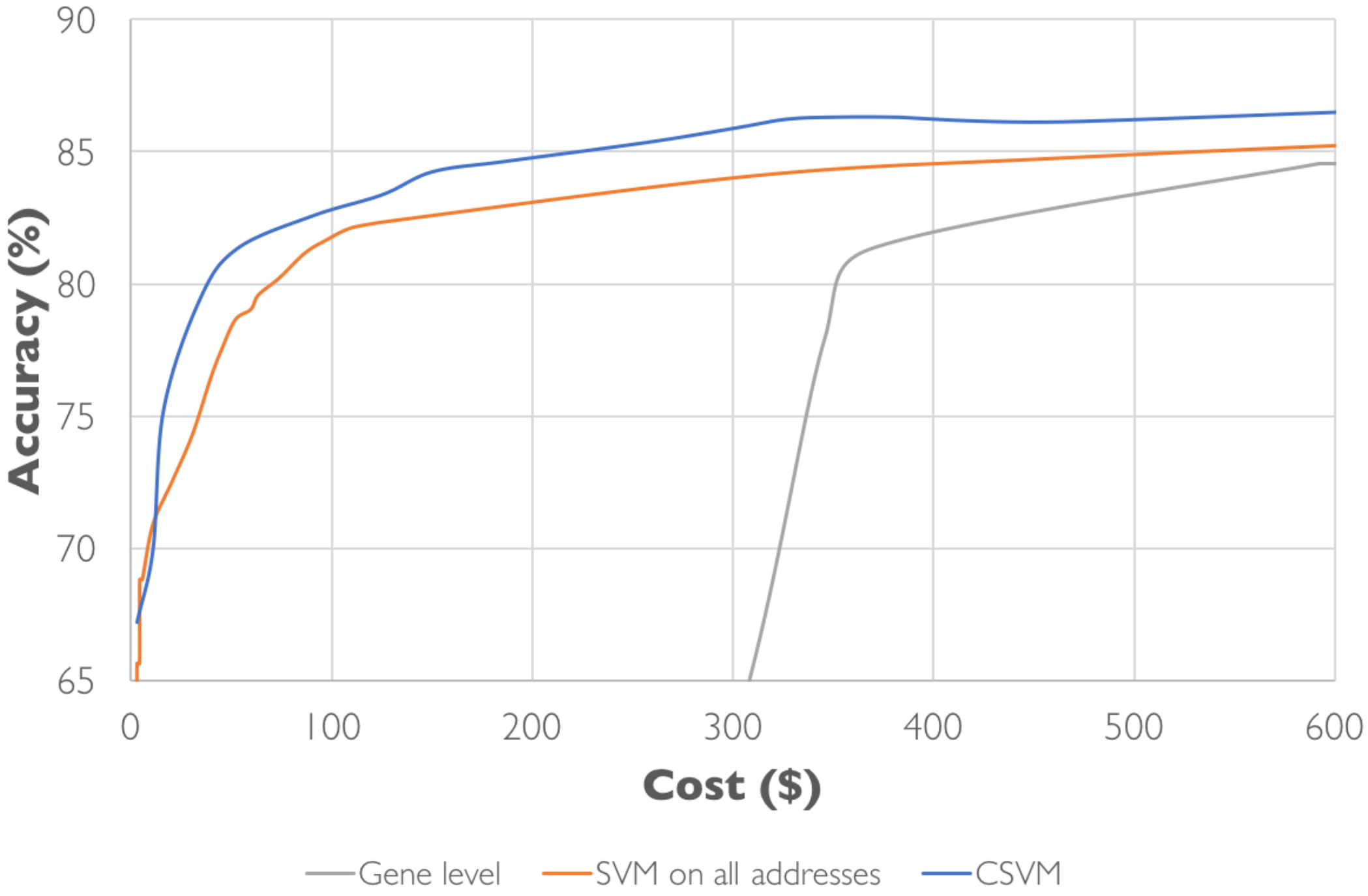
- $t_i = 1$ if sample i is covered
- $u_a = 1$ if address a is chosen
- #variables = #samples + #addresses
  - Significant reduction in problem size
- Column generation problem is a generalization of this problem and yields a tractable IP over a gene

Column generation

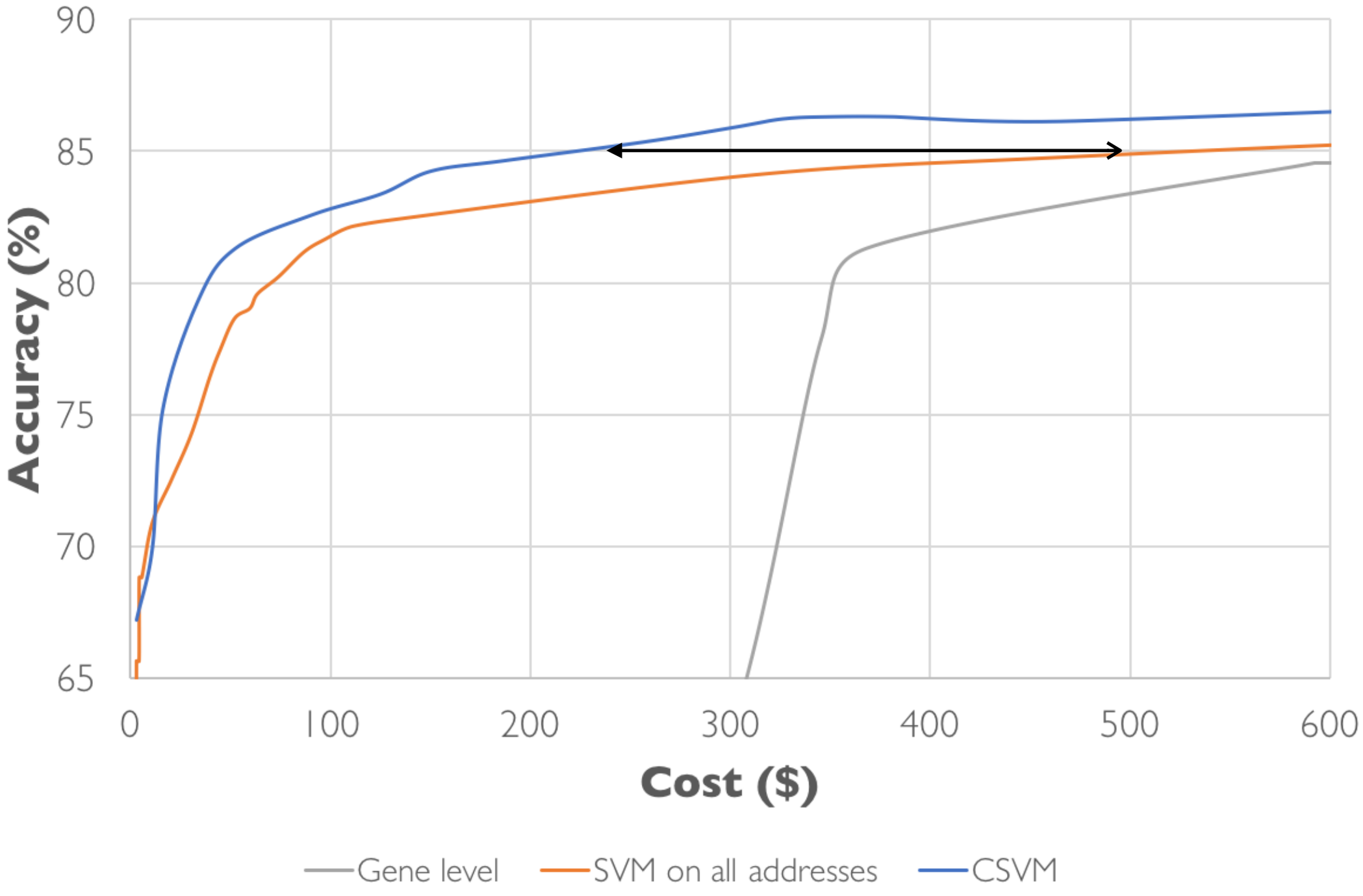
$$\max e^T t$$
$$\text{s.t.} \sum_{a \in A_i} u_a \geq t_i \quad \forall i \in [m]$$
$$u, t \in \{0, 1\}$$

- $t_i = 1$ if sample i is covered
- $u_a = 1$ if address a is chosen
- #variables = #samples + #addresses
  - Significant reduction in problem size
- Column generation problem is a generalization of this problem and yields a tractable IP over a gene
  - ~ 0.2 s to solve

Performance

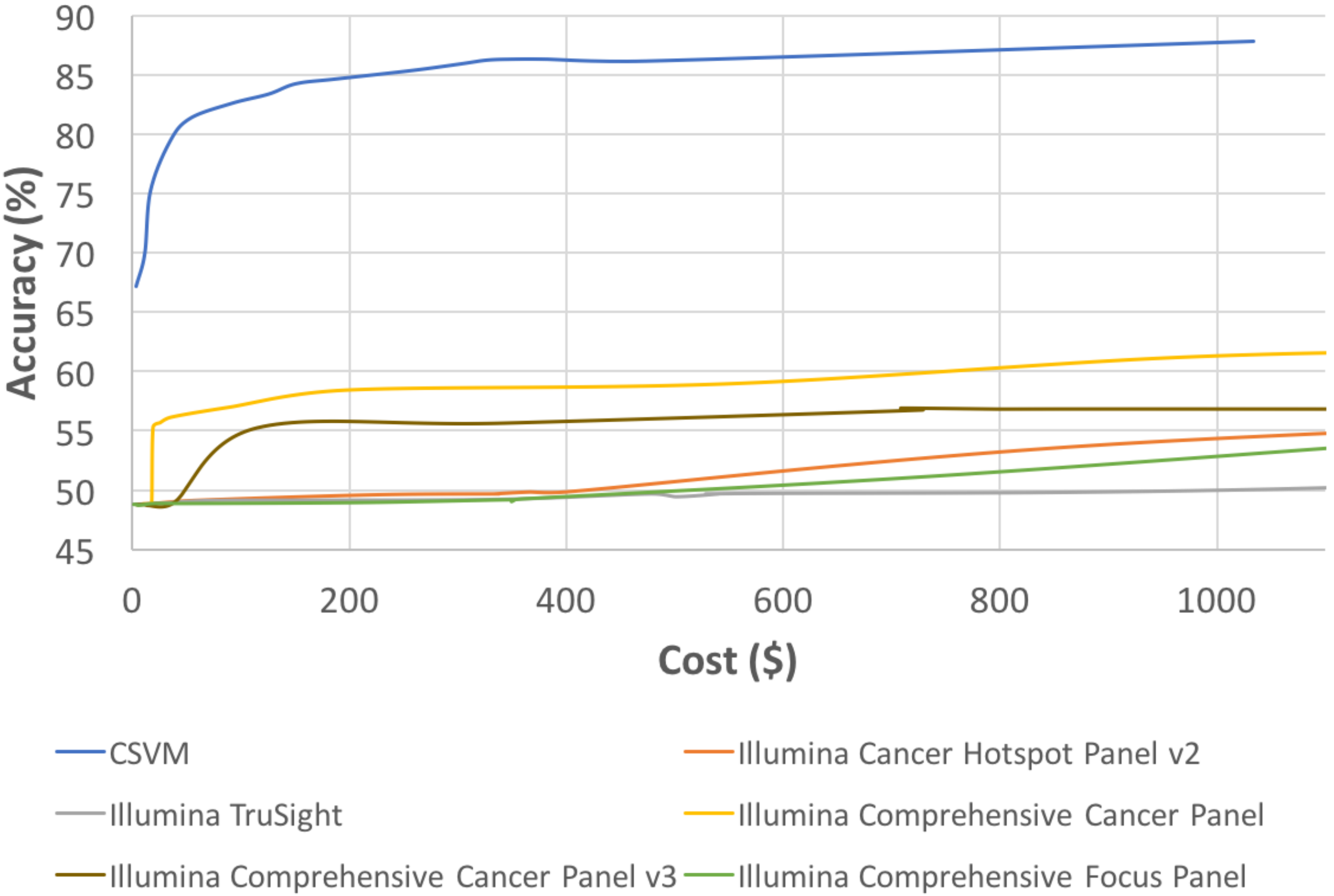


Performance



CSVM achieves a $420 cost reduction for attaining accuracy of 85%

Performance



Summary

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test
- To make a cost-effective test, we need to sequence small parts of the DNA which effectively differentiate between the two cancer types

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test
- To make a cost-effective test, we need to sequence small parts of the DNA which effectively differentiate between the two cancer types
- We propose the CSVM model to solve this problem:

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test
- To make a cost-effective test, we need to sequence small parts of the DNA which effectively differentiate between the two cancer types
- We propose the CSVM model to solve this problem:
  - Incorporates the fixed and variable component of sequencing cost

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test
- To make a cost-effective test, we need to sequence small parts of the DNA which effectively differentiate between the two cancer types
- We propose the CSVM model to solve this problem:
  - Incorporates the fixed and variable component of sequencing cost
  - Effectively solves the classification problem over a large number of features

Summary

- Tremendous reductions in DNA sequencing cost has made cancer detection possible through a blood test
- To make a cost-effective test, we need to sequence small parts of the DNA which effectively differentiate between the two cancer types
- We propose the CSVM model to solve this problem:
  - Incorporates the fixed and variable component of sequencing cost
  - Effectively solves the classification problem over a large number of features
- Illustrate superior performance of CSVM over benchmark algorithms and existing commercial panels

Thanks!

Set cover

Set cover

- For a gene, generalization of set cover problem

Set cover

- For a gene, generalization of set cover problem

$$\begin{aligned} \min\ & c^T u \\ \text{s.t.}\ & \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m] \\ & e^T t > \lambda \\ & u, t \in \{0,1\} \end{aligned}$$

Set cover

- For a gene, generalization of set cover problem

$$\min\ c^T u$$

$$\text{s.t.} \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m]$$

$$e^T t > \lambda$$

$$u, t \in \{0, 1\}$$

Minimize sum of costs of chosen features

Set cover

- For a gene, generalization of set cover problem

$$\min\ c^T u$$

$$\text{s.t.} \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m]$$

$$e^T t > \lambda$$

$$u, t \in \{0, 1\}$$

i-th sample contains mutations in intervals S_i

It is 'covered' if any of these intervals are chosen

Set cover

- For a gene, generalization of set cover problem

$$\min\ c^T u$$

$$\text{s.t.}\ \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m]$$

$$\boxed{e^T t \geq \lambda}$$

$$u, t \in \{0, 1\}$$

samples covered should be at least λ

Set cover

- For a gene, generalization of set cover problem

$$\min\ c^T u$$

$$\text{s.t.}\ \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m]$$

$$\boxed{\left|y^T t\right| \geq \lambda}$$

$$u, t \in \{0, 1\}$$

imbalance should be at least λ

Set cover

- For a gene, generalization of set cover problem

$$\min\ c^T u$$
$$\text{s.t.}\ \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m]$$
$$\left|y^T t\right| \geq \lambda$$
$$u, t \in \{0, 1\}$$

Set cover

- For a gene, generalization of set cover problem

$$\begin{aligned} &\min\ c^T u \\ &\text{s.t.}\ \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m] \\ &\left|y^T t\right| \geq \lambda \\ &u, t \in \{0, 1\} \end{aligned}$$

- Tractable integer program over a gene
  - ~ 0.2 s to solve

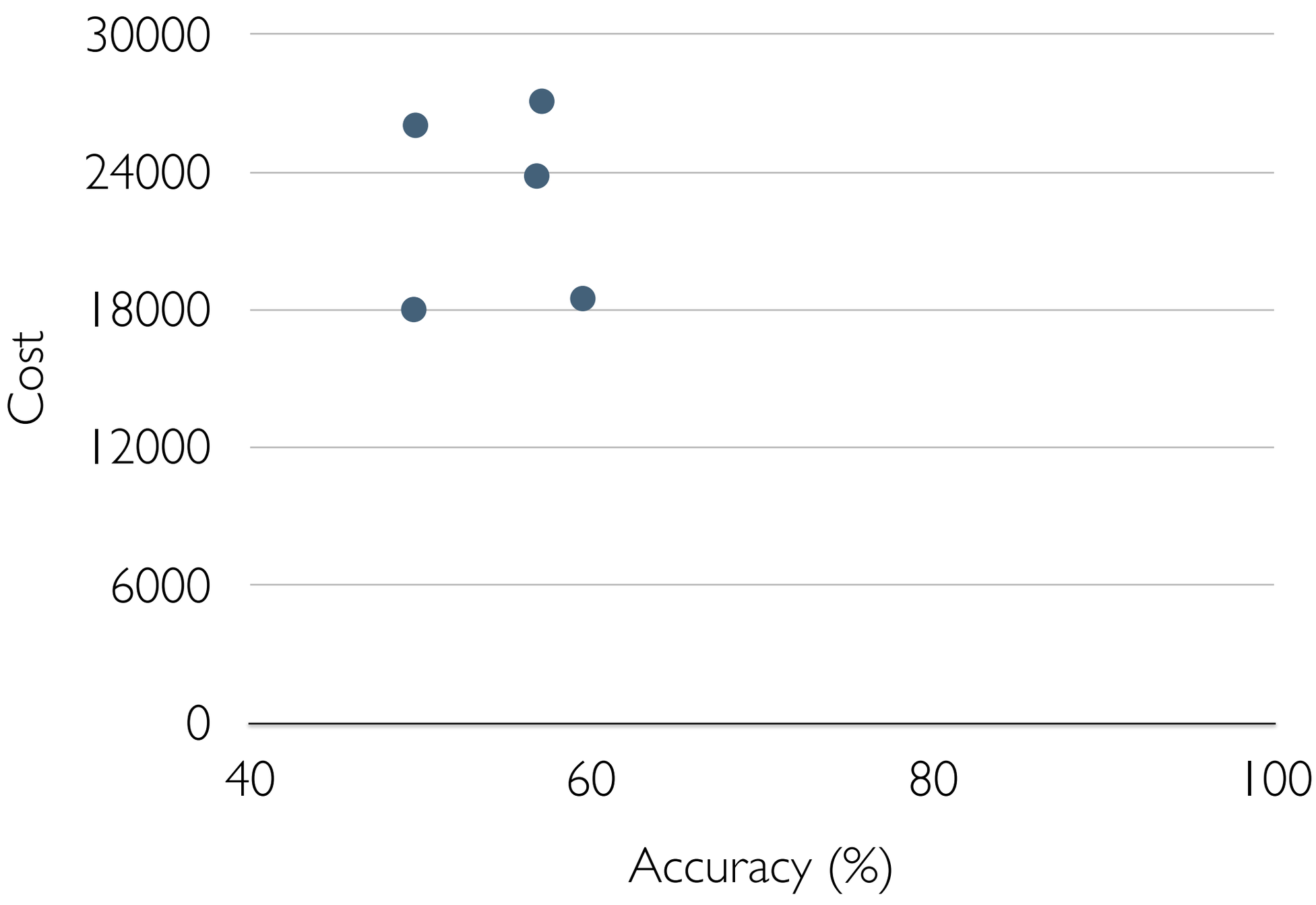
Set cover

- For a gene, generalization of set cover problem

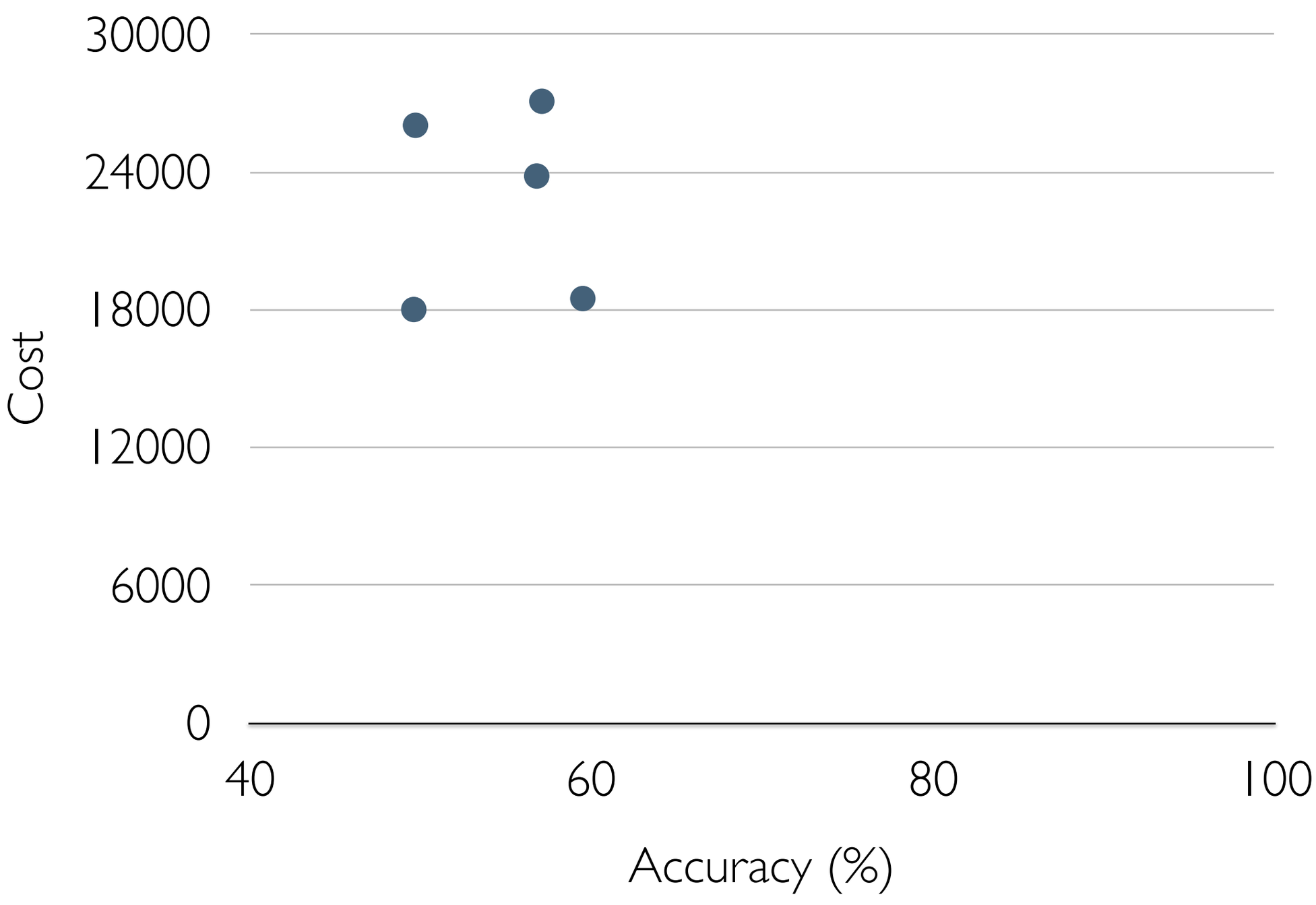
$$\begin{aligned} &\min\ c^T u \\ &\text{s.t.}\ \sum_{s \in S_i} u_s \geq t_i \quad \forall i \in [m] \\ &\left|y^T t\right| \geq \lambda \\ &u, t \in \{0, 1\} \end{aligned}$$

- Tractable integer program over a gene
  - ~ 0.2 s to solve
- Sort genes by imbalance to cost ratio of 'interval' features
- Solve over genes successively until a relevant feature is discovered

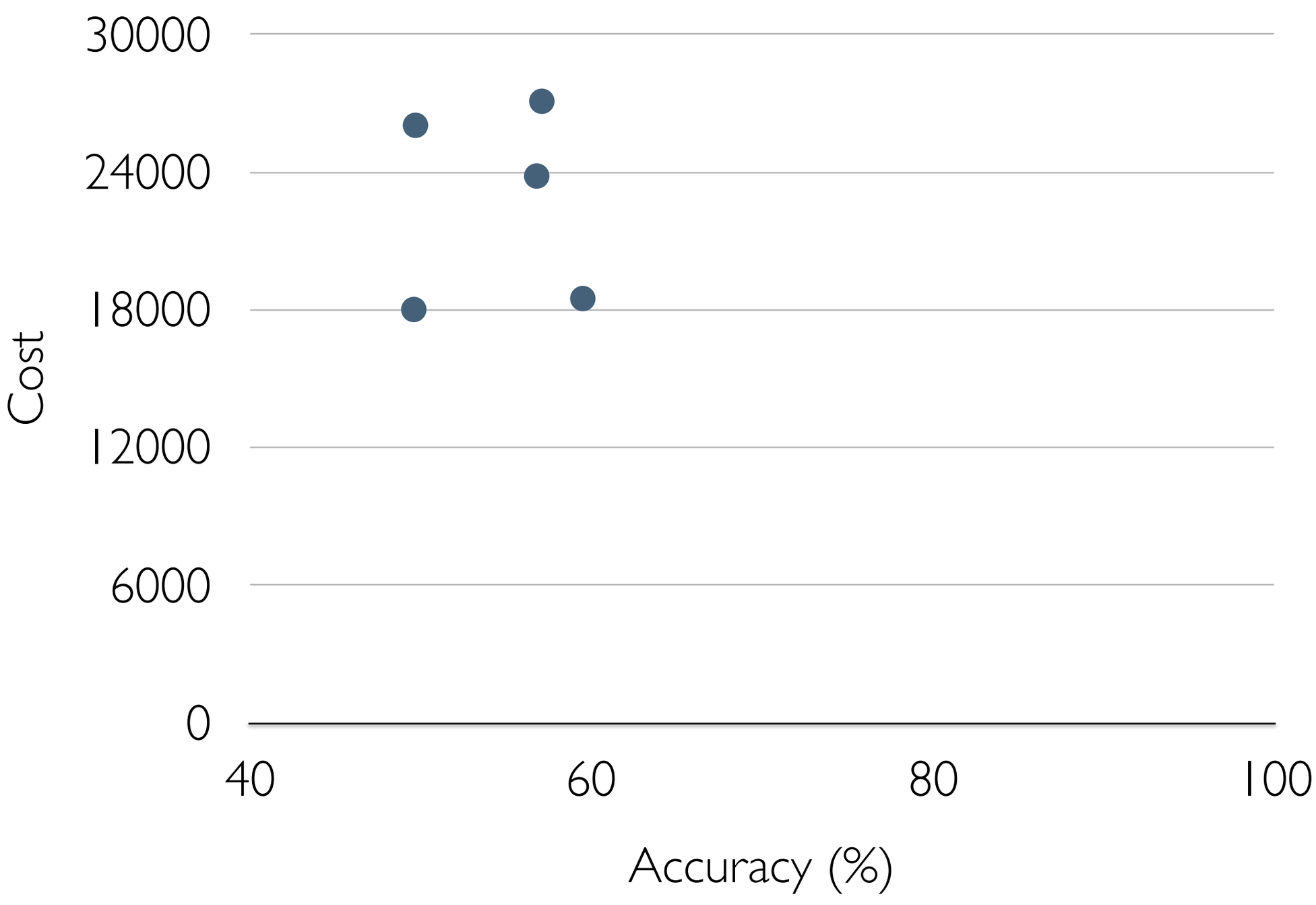
Performance of commercial panels



Performance of commercial panels



Performance of commercial panels



$$\min_{a,b} \sum_{i=1}^{m} [1 - y_i(a^T x_i + b)]^+$$

$$\text{s.t. cost of classifier} \leq \text{budget}$$